



Quantum Algorithm for Molecular Properties and Geometry Optimization

Citation

Kassal, Ivan, and Alán Aspuru-Guzik. 2009. Quantum algorithm for molecular properties and geometry optimization. *Journal of Chemical Physics* 131(22): 224102.

Published Version

doi:10.1063/1.3266959

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:4685247>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

Quantum Algorithm for Molecular Properties and Geometry Optimization

Ivan Kassal* and Alán Aspuru-Guzik†

Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138

(Dated: August 13, 2009)

It is known that quantum computers, if available, would allow an exponential decrease in the computational cost of quantum simulations. We extend this result to show that the computation of molecular properties (energy derivatives) could also be sped up using quantum computers. We provide a quantum algorithm for the numerical evaluation of molecular properties, whose time cost is a constant multiple of the time needed to compute the molecular energy, regardless of the size of the system. Molecular properties computed with the proposed approach could also be used for the optimization of molecular geometries or other properties. For that purpose, we discuss the benefits of quantum techniques for Newton's method and Householder methods. Finally, global minima for the proposed optimizations can be found using the quantum basin hopper algorithm, which offers an additional quadratic reduction in cost over classical multi-start techniques.

Applying *ab initio* methods of quantum chemistry to particular problems often requires computing derivatives of the molecular energy. For instance, obtaining a molecule's electric properties relies on the ability to compute derivatives with respect to external electromagnetic fields. Likewise, computing the gradient of the molecular energy with respect to the nuclear coordinates is the most commonly used method for the proper characterization of potential energy surfaces and for optimizing the geometry of all but the smallest molecules. The computation of these kinds of derivatives, known as *molecular properties*, is nowadays a routine matter when it comes to low-order derivatives or small systems (or both). This is largely due to advances in analytical gradient techniques, which allow for explicit property evaluation without resorting to numerical differentiation [1, 2, 3, 4, 5, 6, 7].

Nevertheless, the computation of higher-order derivatives is often prohibitively expensive, even though such derivatives are often needed. For example, third- and fourth-order anharmonic constants are sometimes required to accurately compute a vibrational absorption spectrum [3] or efficiently determine the location of transition states on complex potential energy surfaces [6]. Other properties of interest, such as hyperpolarizabilities, Raman intensities, or vibrational circular dichroism, are also cubic or quartic derivatives. In this report, we show that quantum computers, once available, will be able to bypass some of the high cost of computing these properties. In particular, we show that any molecular property can be evaluated on a quantum computer using resources that, up to a small constant, are equal to those required to compute the molecular energy once. We have previously characterized the advantage of quantum computers at both computing molecular energies [8, 9] and simulating chemical reaction dynamics [10], and the present work extends our program to molecular properties.

This paper begins with a brief overview of classical

techniques for the evaluation of molecular properties, both numerical and analytical. We then introduce the quantum algorithm for molecular properties, and discuss its advantages and disadvantages with respect to classical techniques. We conclude with geometry optimization as a particular example, and we show that it can benefit from an additional quadratic speed-up through Grover's search [11].

I. THE CLASSICAL METHODS

Given an external perturbation μ , the total molecular electronic energy can be expanded in a Taylor series

$$E(\mu) = E^{(0)} + \mu^\top \mathbf{E}^{(1)} + \frac{1}{2} \mu^\top \mathbf{E}^{(2)} \mu + \dots \quad (1)$$

where the coefficients $\mathbf{E}^{(n)}$ are called the *molecular properties* and describe the response of the system to the applied perturbation [7]. We consider time-independent properties, which can be obtained by differentiating the energy at $\mu = \mathbf{0}$,

$$\mathbf{E}^{(n)} = \left. \frac{d^n E}{d\mu^n} \right|_0. \quad (2)$$

Many examples of useful derivatives can be given. For instance, the derivatives with respect to the electric field \mathbf{F} are the permanent electric dipole, the static polarizability, and the static hyperpolarizabilities of various orders:

$$\left. \frac{dE}{d\mathbf{F}} \right|_0 = -\mathbf{d}, \quad \left. \frac{d^2 E}{d\mathbf{F}^2} \right|_0 = -\boldsymbol{\alpha}, \quad \left. \frac{d^3 E}{d\mathbf{F}^3} \right|_0 = -\boldsymbol{\beta}, \dots \quad (3)$$

where the subscript denotes differentiation at $\mathbf{F} = \mathbf{0}$. The derivatives with respect to nuclear coordinates \mathbf{R} include the forces on the nuclei and the force constants, while mixed derivatives can provide information such as Raman intensities [2].

On a classical computer, an energy derivative can be evaluated either numerically or analytically, and we discuss each approach in turn.

*Electronic address: kassal@fas.harvard.edu

†Electronic address: aspuru@chemistry.harvard.edu

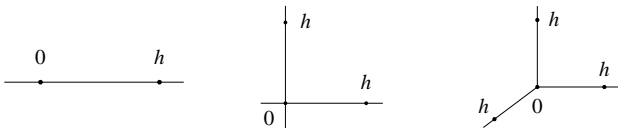


Figure 1: Obtaining a numerical gradient of a function defined on a d -dimensional space classically requires sampling the function $d + 1$ times, once at the origin and once at a distance h along each of the axes. Shown above are the sample points for the cases $d = 1$ through $d = 3$. The quantum gradient algorithm can evaluate many sample points in superposition, producing the same calculated gradient using only one call to the function.

Numerical derivative techniques rely on computing the value of the energy at several discrete points, and then using those values to estimate the true derivative. The simplest technique is finite difference, which for the first derivative in one dimension is the familiar formula,

$$\left. \frac{dE}{d\mu} \right|_0 \approx \frac{E(h) - E(0)}{h}. \quad (4)$$

In d dimensions, computing the gradient requires at least $d+1$ evaluations of the energy, once at the origin and once at a distance h along each axis (Fig. 1). Similarly, evaluating higher-order derivatives requires the knowledge of the energy on a particular grid, with at least $d^n + 1$ points for the n^{th} derivative.

While numerical gradient techniques usually require minimal effort to implement, they are occasionally susceptible to numerical instability, due to the ill-posedness of numerical differentiation in general [12]. This is particularly problematic when using finite-precision arithmetic, where various rounding errors can accumulate and be amplified upon division by the small number h . The fact that small errors in the evaluated function can lead to large errors in the derivative affects *ab initio* electronic structure methods insofar as they usually involve long calculations with many potential sources of error, including rounding and quadrature.

By contrast, analytic derivative techniques are those that compute the derivative by direct evaluation of an analytic expression. They were introduced in quantum chemistry by Pulay [1], and have since largely supplanted numerical procedures. They are numerically more stable and, more importantly, they are usually faster as well.

Analytic gradient formulas exist for just about all electronic structure techniques and for most kinds of perturbations. To illustrate the argument and establish the correct scaling, we will describe the particularly simple case of derivatives of fully variational wavefunctions. We start by writing the molecular energy as a function $E(\mu, \lambda(\mu))$ of the external perturbation μ and the wavefunction parameters $\lambda(\mu)$. These parameters, such as the configuration interaction coefficients, completely describe the electronic wavefunction. Although $\lambda(\mu)$ is a function of μ , for simplicity we will write only λ . The energy is

said to be *fully variational* with respect to λ if, for any given μ , λ assumes the value λ^* such that the variational condition holds:

$$\left. \frac{\partial E(\mu, \lambda)}{\partial \lambda} \right|_* = 0, \quad (5)$$

where $*$ indicates $\lambda = \lambda^*$. In that case we can write $E(\mu) = E(\mu, \lambda^*)$.

For fully variational wavefunctions, the gradient with respect to μ is given by

$$\begin{aligned} \frac{dE(\mu)}{d\mu} &= \left. \frac{\partial E(\mu, \lambda)}{\partial \mu} \right|_* + \left. \frac{\partial E(\mu, \lambda)}{\partial \lambda} \right|_* \frac{\partial \lambda}{\partial \mu} = \\ &= \left. \frac{\partial E(\mu, \lambda)}{\partial \mu} \right|_* = \left\langle \lambda^* \left| \frac{\partial H}{\partial \mu} \right| \lambda^* \right\rangle \end{aligned} \quad (6)$$

where we have used the variational condition and the Hellman-Feynman theorem. Since one need not know the first-order wavefunction response $\frac{\partial \lambda}{\partial \mu}$, computing the gradient is, to within a small constant factor [6], as hard as computing the energy. That is, once $|\lambda^*\rangle$ is available, calculating the expectation value of the Hamiltonian has approximately the same computational cost as calculating its derivative. However, computing the second derivative matrix does require the knowledge of the first-order wavefunction response. In fact, as a direct consequence of Wigner's $2n + 1$ rule of perturbation theory, one needs to know the first n wavefunction responses in order to calculate the $(2n + 1)^{\text{th}}$ derivative. Computing the responses often becomes the bottleneck, and it is what leads to a higher asymptotic cost of higher-order derivatives. While the gradient requires about the same resources as the energy, the second and third derivatives require resources that scale as $O(d)$ times the cost of computing the energy (where d is the number of degrees of freedom, i.e., the dimension of μ) [6]. This scaling comes about because $O(d)$ time is required to compute the matrix $\frac{\partial \lambda}{\partial \mu}$. Likewise, the scaling of the $(2n + 1)^{\text{th}}$ derivative is $O(d^n)$, because the bottleneck becomes the computation of the n^{th} order wavefunction response. In other words, the computational cost of finding the n^{th} analytical derivative is $O(d^{\lceil n/2 \rceil})$, roughly a quadratic speed-up over the $O(d^n)$ numerical methods of the same degree.

The fact that the scaling of derivative techniques, both numerical and analytical, depends on d has meant that these techniques are often restricted to small systems [29]. This is most acutely true of the molecular Hessian, which is often beyond reach, even though the gradient is routinely accessible. We now show that if quantum computers were available, the cost of the higher derivatives would no longer be prohibitive.

II. THE QUANTUM ALGORITHM

The quantum algorithm for molecular properties is based on Jordan's quantum gradient estimation algorithm [13]. Jordan's method can numerically compute

	Classical	Quantum
Derivative	Numerical	Analytical
$\frac{dE}{d\boldsymbol{\mu}}$	$d + 1$	$O(1)$
$\frac{d^2E}{d\boldsymbol{\mu}^2}$	$d^2 + 1$	$O(d)$
$\frac{d^3E}{d\boldsymbol{\mu}^3}$	$d^3 + 1$	$O(d)$
\vdots	\vdots	\vdots
$\frac{d^n E}{d\boldsymbol{\mu}^n}$	$d^n + 1$	$O(d^{\lfloor n/2 \rfloor})$
		2^{n-1}

Table I: Time resources required by various techniques of computing molecular properties, in terms of the cost of computing the energy. For example, the entry “ $d + 1$ ” means that computing the property requires $d + 1$ evaluations of the molecular energy, while the entries in the “Analytical” column indicate comparable computational effort. E is the total electronic energy, $\boldsymbol{\mu}$ is the external perturbation, and d is the dimension of $\boldsymbol{\mu}$. All the derivatives are evaluated at $\boldsymbol{\mu} = \mathbf{0}$. On classical computers, the numerical scalings correspond to the simplest finite-difference scheme. Analytical techniques are the ones that evaluate the derivative directly (the exponent $\lfloor n/2 \rfloor$ comes from Wigner’s $2n + 1$ rule). On a quantum computer, the quantum gradient estimation algorithm is used. It should be noted that on a quantum computer, the number of evaluations of E needed for any derivative is independent of d , and thus of the size of the system.

the gradient of any function F , given a black box (oracle) that computes the value of F for an arbitrary input. In particular, the algorithm can evaluate the gradient using a *single* query to F , regardless of the number of dimensions d of the domain of F . By contrast, the simplest classical finite-difference scheme would require $d + 1$ queries to F (see Fig. 1). The speed-up is essentially achieved by being able to sample along all the d dimensions in superposition. We apply Jordan’s algorithm to the computation of molecular properties by specifying a way to compute the energy on a quantum computer as well as by outlining how to obtain higher derivatives. In this section, we describe the algorithm, its application to quantum chemistry, and finally argue that a return to numerical techniques for molecular properties would be justified if quantum computers became feasible.

We assume that the molecular energy is a smooth, bounded function of the perturbation, $E : [-\frac{h}{2}, \frac{h}{2}]^d \rightarrow [E_{\min}, E_{\max}]$, where a small h is chosen so that E varies sufficiently slowly over the domain. For convenience, we express the perturbations in units such that h is unitless and such that the bounds are the same along all of the axes. We also assume that we have a black box for E , which, given a perturbation $\boldsymbol{\mu}$, outputs the energy $E(\boldsymbol{\mu})$. The precise nature of the algorithm inside the black box is not important, so long as it can be implemented on a quantum computer. In particular, any classical technique of electronic-structure theory can be converted into a quantum algorithm [14]. In Sec. III, we will discuss possible realizations of the black box, including the use of quantum simulation algorithms, which offer a signifi-

cant improvement over classical ones.

We begin by choosing the number n of bits of precision that we desire in the final gradient. Jordan’s algorithm starts in an equal superposition on d registers of n qubits each (nd qubits total) [14]:

$$\frac{1}{\sqrt{N^d}} \sum_{k_1=0}^{N-1} \cdots \sum_{k_d=0}^{N-1} |k_1\rangle \cdots |k_d\rangle = \frac{1}{\sqrt{N^d}} \sum_{\mathbf{k}} |\mathbf{k}\rangle, \quad (7)$$

where $N = 2^n$, the states $|k_i\rangle$ are integers on n qubits represented in binary, and $|\mathbf{k}\rangle$ is a d -dimensional vector of all the $|k_i\rangle$ ’s.

We use this state as an input for the black box for E , which will, for every integer-vector $|\mathbf{k}\rangle$ in the superposition, append a phase proportional to the energy $E(\boldsymbol{\mu})$ at perturbation $\boldsymbol{\mu} = h(\mathbf{k} - \mathbf{N}/2)/N$ (where \mathbf{N} is the vector (N, N, \dots, N) and serves to center the sampling region on the origin). To achieve maximum precision with fewest qubits, one needs an estimate m of the largest magnitude of any of the first derivatives of E . Then, the energy evaluated by the black box is scaled by a factor $2\pi N/hm$. Because the black box operates on all the terms in the superposition at once, a single call results in the state

$$\begin{aligned} & \frac{1}{\sqrt{N^d}} \sum_{\mathbf{k}} \exp \left[\frac{2\pi i N}{hm} E \left(\frac{h}{N} (\mathbf{k} - \mathbf{N}/2) \right) \right] |\mathbf{k}\rangle \approx \\ & \approx \frac{1}{\sqrt{N^d}} \sum_{\mathbf{k}} \exp \left[\frac{2\pi i N}{hm} \left(E(\mathbf{0}) + \frac{h}{N} (\mathbf{k} - \mathbf{N}/2) \cdot \left. \frac{dE}{d\boldsymbol{\mu}} \right|_{\mathbf{0}} \right) \right] |\mathbf{k}\rangle. \end{aligned} \quad (8)$$

The neglect of terms quadratic in h and higher is a valid approximation for sufficiently small h (the error caused by higher-order terms is discussed in [13] and is only polynomial). The final state is separable and equals

$$\begin{aligned} & \frac{e^{i\Phi(\mathbf{0})}}{\sqrt{N^d}} \sum_{k_1=0}^{N-1} \exp \left[\frac{2\pi i}{m} k_1 \left. \frac{\partial E}{\partial \mu_1} \right|_{\mathbf{0}} \right] |k_1\rangle \cdots \\ & \cdots \sum_{k_d=0}^{N-1} \exp \left[\frac{2\pi i}{m} k_d \left. \frac{\partial E}{\partial \mu_d} \right|_{\mathbf{0}} \right] |k_d\rangle, \end{aligned} \quad (9)$$

with phase

$$\Phi(\mathbf{0}) = 2\pi \left(\frac{N}{hm} E(\mathbf{0}) - \frac{\mathbf{N}}{2m} \cdot \left. \frac{dE}{d\boldsymbol{\mu}} \right|_{\mathbf{0}} \right). \quad (10)$$

Applying the inverse quantum Fourier transform [14] to each of the d registers results in the gradient

$$e^{i\Phi(\mathbf{0})} \left| \frac{N}{m} \left. \frac{\partial E}{\partial \mu_1} \right|_{\mathbf{0}} \right\rangle \cdots \left| \frac{N}{m} \left. \frac{\partial E}{\partial \mu_d} \right|_{\mathbf{0}} \right\rangle = e^{i\Phi(\mathbf{0})} \left| \frac{N}{m} \left. \frac{dE}{d\boldsymbol{\mu}} \right|_{\mathbf{0}} \right\rangle. \quad (11)$$

The scaling factor N/m ensures that $\frac{N}{m} \frac{dE}{d\boldsymbol{\mu}}$ is an integer-vector with n bits of precision along each dimension. It should be reiterated that a single call to E was made, as opposed to the $d + 1$ that would be needed in the case of numerical differentiation by finite difference.

Overall, the gradient estimation algorithm produces the transformation

$$|0\rangle \longrightarrow e^{i\Phi(\mathbf{0})} \left| \frac{N}{m} \frac{dE}{d\boldsymbol{\mu}} \right|_{\mathbf{0}} \rangle. \quad (12)$$

We can compute the Hessian (and higher derivatives) by iterating this algorithm. If, instead of making a call to $E(\boldsymbol{\mu})$, the algorithm sought $E(\boldsymbol{\mu}-\boldsymbol{\nu})$ from the black box, we would perform, at the cost of this single additional subtraction,

$$|0\rangle \longrightarrow e^{i\Phi(\boldsymbol{\nu})} \left| \frac{N}{m} \frac{dE}{d\boldsymbol{\mu}} \right|_{\boldsymbol{\nu}} \rangle, \quad (13)$$

with global phase

$$\Phi(\boldsymbol{\nu}) = 2\pi \left(\frac{N}{hm} E(\boldsymbol{\nu}) - \frac{\mathbf{N}}{2m} \cdot \frac{dE}{d\boldsymbol{\mu}} \right|_{\boldsymbol{\nu}} \rangle. \quad (14)$$

Because we will be using this procedure as a subroutine, it is important to remove (or “uncompute”) the global phase, which would otherwise become a relative phase. One additional evaluation of E (at $\boldsymbol{\nu}$) suffices for this uncomputation. Overall, this supplies another black box, which, given $\boldsymbol{\nu}$, yields $\left| \frac{N}{m} \frac{dE}{d\boldsymbol{\mu}} \right|_{\boldsymbol{\nu}} \rangle$ using only two calls to the original black box for E . One can use the gradient algorithm with this new black box, producing the state

$$e^{i\Phi^{(2)}(\mathbf{0})} \left| \frac{N}{m^{(2)}} \frac{d^2E}{d\boldsymbol{\mu}^2} \right|_{\mathbf{0}} \rangle, \quad (15)$$

which is a two-dimensional array of d^2 quantum registers containing all the elements of the Hessian matrix of E . In addition, $m^{(2)}$ is an estimate for the magnitude of the largest second derivative, and the phase is

$$\Phi^{(2)}(\mathbf{0}) = 2\pi \left(\frac{N}{hm^{(2)}} \frac{dE}{d\boldsymbol{\mu}} \right|_{\mathbf{0}} - \frac{\mathbf{N}}{2m^{(2)}} \cdot \frac{d^2E}{d\boldsymbol{\mu}^2} \right|_{\mathbf{0}} \rangle. \quad (16)$$

Computing higher derivatives would require additional factors of two in the number of required black box calls, caused by the need to uncompute a global phase at each step (this problem is a common feature when it comes to recursing quantum algorithms [15]). Hence, evaluating the n^{th} derivative requires 2^{n-1} queries to E , which, although exponential in n , is much better than $d^n + 1$, which is the minimum number of function queries required to compute the derivative by classical finite difference. We stress that the number of calls to E is independent of d , and thus of the size of the system, for the derivative of any order.

One could remark that the quantum gradient algorithm is a numerical approach and that therefore, just like classical numerical techniques, it would be affected by numerical instability. This implies that the quantum gradient algorithm cannot be used indiscriminately for problems that feature errors that cannot be controllably

reduced through additional computational effort. For example, finite-precision arithmetic presents the same problems to quantum computers as it does to classical ones, but the rounding errors can be brought under control by using more digits of precision (as on classical computers). Quantum chemistry techniques might present numerical problems as well, insofar as they contain uncontrolled sources of error. However, if the technique for computing the energy is numerically exact, that is to say, if the error in the energy can be controllably reduced below any level, the magnitude of the numerical error in the calculated derivative can likewise be made arbitrarily small. Fortunately, quantum computers would make it possible to efficiently evaluate the numerically exact molecular energy, meaning that numerical instability will not be a problem. We turn to the topic of molecular energy evaluation next.

III. THE BLACK BOX FOR THE ENERGY

The application of Jordan’s gradient algorithm to chemical problems requires that there be a black box that can compute the value of the ground-state molecular energy at any value of the perturbation $\boldsymbol{\mu}$ in the neighborhood of $\boldsymbol{\mu} = \mathbf{0}$. Furthermore, to avoid numerical artifacts, this black box should be numerically exact, allowing the error in the energy to be controllably reduced through additional computational work.

The problem of exact classical electronic structure methods is that they generally have a computational cost that scales exponentially with the size of the system. Although these classical algorithms could also be used as subroutines in the quantum gradient algorithm, there are quantum electronic-structure algorithms that could avoid the exponential scaling in many cases.

In particular, we have recently described a quantum full CI algorithm [8] for computing the molecular ground state energy to a given precision in $O(M^5)$ time [16], where M is the number of basis functions. This algorithm could be easily recast as a subroutine that would function as the black box for the energy. Several modifications would have to be made, including the direct computation of the overlap integrals on the quantum computer, rendering it possible to introduce the perturbation $\boldsymbol{\mu}$ into the calculation. Nevertheless, a quantum computer running the quantum FCI algorithm could be used to obtain a molecular property of a system with basis size M in $O(M^5)$ time, a dramatic improvement over the possibilities of classical computers.

A more recent development is the real-space chemical dynamics simulation algorithm [10, 17], based on Zalka’s earlier work [18]. It is known that simulating, to a given precision, the exact dynamics of a system of P particles interacting under a pairwise interaction requires at most $O(P^2)$ time and $O(P)$ space, in contrast to the classical exponential cost. If an eigenstate of the system Hamiltonian were prepared as the initial state [19],

the dynamics would only apply a phase to the wavefunction. This phase could be read out by the phase estimation algorithm [14, 20, 21], forming the required energy black box. Although the large prefactor of this algorithm would make it slower for small molecules than the equivalent quantum FCI calculation, it benefits from a superior asymptotic scaling as well as from the fact that only minimal modifications would need to be made to insert the perturbation μ into the calculation. For example, simulations with different nuclear coordinates proceed in exactly the same way, while an electromagnetic field requires only a small modification of the simulated Hamiltonian [18].

It should be remarked that current quantum algorithms for energy estimation, such as the ones mentioned above, rely on quantum phase estimation, which has been criticized as inefficient [22] because its cost grows exponentially with the number of bits of precision sought. This could be significant for gradient estimation, which may require precise energy evaluations to avoid numerical errors. To estimate the cost, we note that if the gradient is desired to n bits of precision (as in Eq. 11), the black box should evaluate the energy to

$$n_E = \log_2 \left[\frac{E_{\max} - E_{\min}}{(mh/2^n)(\theta/2\pi)} \right] \approx n + \log_2 \frac{2\pi}{\theta} \quad (17)$$

bits of precision [13], where $\cos^2 \theta$ is the desired success probability of the algorithm. For example, with $\theta = \pi/8$, the algorithm succeeds 85% of the time and requires four more digits of precision in the energy than is desired in the gradient. The four additional digits present only a constant overhead, meaning that the computation of any molecular property at any precision is, up to a constant factor, as hard as computing the energy of the same molecule at the same precision.

Finally, a limitation of current quantum simulation algorithms is that they are generally spin-free and non-relativistic, which limits the ability to compute derivatives such as indirect spin-spin coupling.

IV. NEWTON'S METHOD AND GEOMETRY OPTIMIZATION

Perhaps the single most common use of molecular derivatives is molecular geometry optimization. We can therefore use it to illustrate some of the advantages of a quantum algorithm over a classical one, including a quantum version of Newton's method, which offers an additional quadratic speedup over its classical counterpart.

A simple way for finding the locally optimal geometry is to perform the standard Newton iterations,

$$\mathbf{R}_{n+1} = \mathbf{R}_n - \left(\frac{dE}{d\mathbf{R}} \Big|_{\mathbf{R}_n} \right) \cdot \left(\frac{d^2E}{d\mathbf{R}^2} \Big|_{\mathbf{R}_n} \right)^{-1}, \quad (18)$$

until convergence is reached. Here, \mathbf{R}_n are the nuclear coordinates at the n^{th} iteration, and $\frac{dE}{d\mathbf{R}} \Big|_{\mathbf{R}_n}$ and $\frac{d^2E}{d\mathbf{R}^2} \Big|_{\mathbf{R}_n}$ are, respectively, the gradient and Hessian of E with respect to nuclear displacement (the ‘‘molecular gradient’’ and the ‘‘molecular Hessian’’). If a quantum computer were used to compute the derivatives, one would require exactly 3 calls to a black box for E per iteration: one for the gradient and two for the Hessian. A classical approach, on the other hand, would be much slower, requiring at least $d^2 + 1$ function calls for finite difference, and approximately $O(d)$ effort in the analytical case [30]. For large molecules with large d , this savings could prove significant, even if each energy evaluation takes much longer on a quantum machine than on a classical computer.

There are many classical tricks available for speeding up the convergence of Newton's method if the initial guess is not close to a local minimum, in which case the usual Newton step might be inappropriately large. Techniques such as trust regions and level shifts [23] are still available to quantum computers, or they can be implemented as classical post-processing.

In addition, we remark that Newton's method is the first in the class of Householder methods, which offer a rate of convergence of $\ell + 1$, provided that derivatives up to order $\ell + 1$ exist and can be calculated. A quantum computer could be used to accelerate Householder methods of any degree, requiring $\sum_{m=1}^{\ell+1} 2^{m-1} = 2^{\ell+1} - 1$ calls to the black box for order- ℓ Householder optimization method. Although exponential in ℓ , this expression is independent of system dimension d .

Of course, Newton's method is only useful for local minimization, and we are often interested in global optimization. Here, we can use a quantum version of the multistart technique, called the quantum basin hopper [24, 25, 26, 27]. A number of points is selected at random, and each is followed, using a local search, to its local basin (if a quantum version of Newton's method is used for the local search, such as the one we propose above, we can get the usual quadratic convergence). Then, the minima of all the basins are compared and the least one chosen as the global minimum. Quantum computers could add a quadratic speed-up to such a multistart technique, since the resulting local minima form an unstructured database that can be searched using Grover's algorithm [11, 14] with a quadratic speed-up. As Dür and Høyer pointed out [28], a Grover search can find the minimum of an unstructured database with $O(\sqrt{K} \log K)$ calls to the database (where K is the number of database entries, i.e. multistart points), as opposed to the classically required $O(K \log K)$ queries.

V. CONCLUSION

We have shown that Jordan's quantum gradient estimation algorithm can be applied to the estimation of time-independent, non-relativistic molecular properties.

Doing so requires a quantum electronic-structure black box, for which known quantum simulations algorithms are well suited. The quantum algorithm offers a speed-up from the classical cost of $O(d^{\lfloor n/2 \rfloor})$ for analytical derivatives to the quantum query complexity of 2^{n-1} . That is, the number of energy calculations required on the quantum computer is independent of d , and thus of the system size, which could offer a significant advantage for the computation of properties of large systems. In particular, it would make the molecular Hessian of any molecule only twice as expensive as its molecular gradient, enabling a fast, local geometry optimization us-

ing Newton's method. Finally, global optimization could combine the local Newton's method with Grover search to offer an additional quadratic speed-up over the classical multi-start technique.

Acknowledgments

We acknowledge support from the Army Research Office under contract W911NF-07-0304. I.K. thanks the Joyce and Zlatko Baloković Scholarship.

-
- [1] P. Pulay, *Mol. Phys.* **17**, 197 (1969).
 - [2] P. Pulay, *Adv. Chem. Phys.* **69**, 241 (1987).
 - [3] T. Helgaker and P. Jørgensen, *Adv. Quantum Chem.* **19**, 183 (1988).
 - [4] Y. Yamaguchi, J. D. Goddard, Y. Osamura, and H. Schaefer, *A New Dimension to Quantum Chemistry: Analytic Derivative Methods in Ab Initio Molecular Electronic Structure Theory* (Oxford University Press, New York, 1994).
 - [5] R. Shepard, in *Modern Electronic Structure Theory*, edited by D. Yarkony (World Scientific, Singapore, 1995), pp. 345–458.
 - [6] P. Pulay, in *Modern Electronic Structure Theory*, edited by D. Yarkony (World Scientific, Singapore, 1995), pp. 1191–1240.
 - [7] T. Helgaker, in *The Encyclopedia of Computational Chemistry*, edited by P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, and P. R. Schreiner (Wiley, New York, 1998), pp. 1157–1169.
 - [8] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, *Science* **309**, 1704 (2005).
 - [9] H. Wang, S. Kais, A. Aspuru-Guzik, and M. R. Hoffmann, *Phys. Chem. Chem. Phys.* **10**, 5388 (2008).
 - [10] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, *Proc. Natl. Acad. Sci.* **105**, 18681 (2008).
 - [11] L. Grover, *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* pp. 212–219 (1996).
 - [12] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems* (Winston, Washington, 1977).
 - [13] S. P. Jordan, *Phys. Rev. Lett.* **95**, 050501 (2005).
 - [14] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, New York, 2000).
 - [15] S. Aaronson, *Quantum Inform. Comput.* **3**, 165 (2003).
 - [16] B. P. Lanyon, J. D. Whitfield, G. G. Gillet, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White, *arXiv:0905.0887* (2009).
 - [17] D. A. Lidar and H. Wang, *Phys. Rev. E* **59**, 2429 (1999).
 - [18] C. Zalka, *Proc. Roy. Soc. A* **454**, 313 (1998).
 - [19] N. J. Ward, I. Kassal, and A. Aspuru-Guzik, *J. Chem. Phys.* **130**, 194105 (2009).
 - [20] A. Y. Kitaev, *quant-ph/9511026* (1995).
 - [21] D. S. Abrams and S. Lloyd, *Phys. Rev. Lett.* **83**, 5162 (1999).
 - [22] K. R. Brown, R. J. Clark, and I. L. Chuang, *Phys. Rev. Lett.* **97**, 050504 (2006).
 - [23] R. Fletcher, *Practical Methods of Optimization* (Wiley, New York, 2000), 2nd ed.
 - [24] D. Bulger, *quant-ph/0507193* (2005).
 - [25] D. Bulger, *J. Optim. Theory Appl.* **133**, 289 (2007).
 - [26] W. P. Baritomp, D. Bulger, and G. Wood, *SIAM J. Optim.* **15**, 1170 (2005).
 - [27] J. Zhu, Z. Huang, and S. Kais, *arXiv:0906.2375* (2009).
 - [28] C. Dürr and P. Hoyer, *quant-ph/9607014* (1996).
 - [29] For some classes of useful properties, d is independent of system size. For example, if the perturbation is the electric field, then $d = 3$, and indeed there are classical techniques for computing electrical properties of large molecules. The quantum speed-up is therefore most pronounced in cases where d varies with system size, as it does whenever there is differentiation with respect to nuclear coordinates.
 - [30] Quasi-Newton methods (such as the Broyden-Fletcher-Goldfarb-Shanno method), or even simpler methods such as simple gradient descent, can remove the need to compute the molecular Hessian at each step, or at all. While such schemes are useful and widely applied, we do not discuss them here because they are typically slower and, for a given number of steps, less accurate than Newton's method. While they offer a classical computational advantage, on a quantum computer that advantage would be erased by the ability to rapidly compute the exact Hessian at each step.